In the Claims:

1. (Original) A method for providing a Web browser running on a computer with HTTP access to a peer server located behind a firewall in a peer-to-peer network, comprising;

(a) providing the peer-to-peer network with a proxy server;

(b) registering an outbound socket connection with the proxy server by the peer server;

(c) in response to the proxy server receiving an HTTP request to access the peer server from the Web browser, translating the HTTP request into a request packet and sending the request packet to the peer server; and

(d) in response to the peer server receiving the request packet, translating the request packet back into the HTTP request and responding to the request, thereby enabling generic web traffic to flow.

2. (Currently Amended) The method of claim 1 wherein the peer server further includes a Web server, step (d) further including the steps of:

(i) responding to request by passing the HTTP request to the Web server;

(ii) receiving an HTTP response from Web server;

(iii) translating HTTP response into a response packet;

(iv) sending the response packet [[to]] from the peer server to the proxy server over the outbound socket connection;

(v) receiving the response packet on the proxy server and translating a response packet back into the HTTP response; and

(vi) sending the HTTP response from the peer server to the Web browser.

3. (Original) The method of claim 2 wherein the peer-to-peer network includes multiple peer servers, and the proxy server is separate and apart from the peer servers.

4. (Original) The method of claim 3 further including the step of: providing each of the peer servers with a peer node, a Web server, and a Web browser.

5. (Original) The method of claim 4 further including step of: providing the peer-to-peer network with a registration server and a DNS server.

6. (Original) The method of claim 5 wherein step (b) further includes the step of: passing a name of the peer server from the peer server to the registration server, and receiving a name and IP address of the proxy server to which it is assigned.

7. (Original) The method of claim 6 wherein step (b) further includes the step of: registering by the peer server, the name of the proxy server, and the IP address of the proxy server with the DNS server.

8. (Original) The method of claim 7 wherein step (b) further includes the step of: after the peer server registers with the proxy server, notifying a user of the computer via e-mail that content exists on the peer server for viewing, and including a URL of the peer server in the e-mail.

9. (Original) The method of claim 8 wherein step (b) further includes the step of: in response to the user clicking on the URL e-mail, the computer contacts the DNS server to determine an identity of the proxy server in which to send the HTTP request.

10. (Original) A method of claim 3 further including the step of: providing the proxy server with a servlet thread, a registration manager, a peer manager, a peer MessageBox, and a peer packet manager thread.

11. (Original) The method of claim 10 wherein step (c) further includes the step of: receiving the HTTP request as a URL by the servlet thread and using the registration manager to determine if the peer server identified in requesting URL is registered with the peer server, and if so, returning the corresponding peer socket.

12. (Original) The method of claim 11 wherein step (c) further includes the step of: creating, by the servlet thread, a peer request packet, and passing the peer request packet to the peer manager.

13. (Original) The method of claim 12 wherein step (c) further includes the step of: providing the peer request packet with a MessageBoxID, an HTTP URL, HTTP headers, and an HTTP Post Data field.

14. (Original) The method of claim 13 wherein step (c) further includes the step of: finding by the peer manager, the socket connection to the peer server, and passing the peer request packet to the peer server.

15. (Original) The method of claim 2 wherein step (d) further includes the step of: breaking the HTTP response into chunks and sending the chunks to the proxy server in successive peer response packets.

16. (Original) The method of claim 15 wherein step (d) further includes the step of: providing the peer server with several threads for handling HTTP requests from the proxy server, and multiplexing responses to those requests over the same response socket back to the proxy server.

17. (Original) A computer-readable medium containing program instructions for providing a Web browser running on a computer with HTTP access to a peer server located behind a firewall in a peer-to-peer network, the program instructions for;

(a) providing the peer-to-peer network with a proxy server;

(b) registering an outbound socket connection with the proxy server by the peer server;

(c) in response to the proxy server receiving an HTTP request to access the peer server from the Web browser, translating the HTTP request into a request packet, and sending the request packet to the peer server; and

(d) in response to the peer server receiving the request packet, translating the request packet back into the HTTP request and responding to the request, thereby enabling generic web traffic to flow.

18. (Currently Amended) The computer-readable medium of claim 17 wherein the peer server further includes a Web server, instruction (d) further including the instructions of:

(i) responding to request by passing the HTTP request to the Web server;

(ii) receiving an HTTP response from Web server;

(iii) translating HTTP response into a response packet;

(iv) sending the response packet [[to]] from the peer server to the proxy server over the outbound socket connection;

(v) receiving the response packet on the proxy server and translating a response packet back into the HTTP response; and

(vi) sending the HTTP response from the peer server to the Web browser.

19. (Original) The computer-readable medium of claim 18 wherein the peer-to-peer network includes multiple peer servers, and the proxy server is separate and apart from the peer servers.

20. (Original) The computer-readable medium of claim 19 further including the instruction of: providing each of the peer servers with a peer node, a Web server, and a Web browser.

21. (Original) The computer-readable medium of claim 20 further including instruction of: providing the peer-to-peer network with a registration server and a DNS server.

22. (Original) The computer-readable medium of claim 21 wherein instruction (b) further includes the instruction of: passing a name of the peer server from the peer server to the registration server, and receiving a name and IP address of the proxy server to which it is assigned.

23. (Original) The computer-readable medium of claim 22 wherein instruction (b) further includes the instruction of: registering by the peer server, the name of the proxy server and the IP address of the proxy server with the DNS server.

24. (Original) The computer-readable medium of claim 23 wherein instruction (b) further includes the instruction of: after the peer server registers with the proxy server, notifying a user of the computer via e-mail that content exists on the peer server for viewing, and including a URL of the peer server in the e-mail.

25. (Original) The computer-readable medium of claim 24 wherein instruction (b) further includes the instruction of: in response to the user clicking on the URL e-mail, the computer contacts the DNS server to determine an identity of the proxy server in which to send the HTTP request.

26. (Original) A computer-readable medium of claim 19 further including instruction of: providing the proxy server with a servlet thread, a registration manager, a peer manager, a peer MessageBox, and a peer packet manager thread.

27. (Original) The computer-readable medium of claim 26 wherein instruction (c) further includes the instruction of: receiving the HTTP request as a URL by the servlet thread and using the registration manager to determine if the peer server identified in requesting URL is registered with the peer server, and if so, returning the corresponding peer socket.

28. (Original) The computer-readable medium of claim 27 wherein instruction (c) further includes the instruction of: creating, by the servlet thread, a peer request packet, and passing the peer request packet to the peer manager.

29. (Original) The computer-readable medium of claim 28 wherein instruction (c) further includes the instruction of: providing the peer request packet with a MessageBoxID, an HTTP URL, an HTTP headers, and an HTTP Post Data field.

30. (Original) The computer-readable medium of claim 29 wherein instruction (c) further includes the instruction of: finding by the peer manager, the socket connection to the peer server, and passing the peer request packet to the peer server.

31. (Original) The computer-readable medium of claim 18 wherein instruction (d) further includes the instruction of: breaking the HTTP response into chunks and sending the chunks to the proxy server in successive peer response packets.

32. (Original) The computer-readable medium of claim 31 wherein instruction (d) further includes the instruction of: providing the peer server with several threads for handling HTTP requests from the proxy server, and multiplexing responses to those requests over the same response socket back to the proxy server.

33. (Original) A method for providing a web browser with HTTP access to a peer server located behind a firewall in a peer-to-peer network, comprising:

(a) registering an outbound socket connection from the peer server to a proxy server;

(b) redirecting all incoming HTTP requests intended for the peer server to the proxy server;

(c) in response to the proxy server receiving one of the redirected HTTP request, finding the socket connection to the peer server, translating the HTTP requests into a multiplexed protocol comprising a request packet, and sending the request packet to the peer server;

(d) in response to the peer node receiving the request packet, demultiplexing the request, translating the request packet back into the original HTTP request, and passing the HTTP request to a local web server;

(e) in response to the peer node receiving an HTTP response from the Web server, translating the HTTP response into a response packet, and sending the response packet to the proxy server over the outbound socket connection; and

(f) in response to the proxy server receiving the response packet from the peer server, translating the response packet back into the HTTP response, and sending the HTTP response to the requesting Web browser.

34. (Original) A computer-readable medium containing program instructions for providing a web browser with HTTP access to a peer server located behind a firewall in a peer-to-peer network, the program instructions for:

(a) registering an outbound socket connection from the peer server to a proxy server;

(b) redirecting all incoming HTTP requests intended for the peer server to the proxy server;

(c) in response to the proxy server receiving one of the redirected HTTP request, finding the socket connection to the peer server, translating the HTTP requests into a multiplexed protocol comprising a request packet, and sending the request packet to the peer server;

(d) in response to the peer node receiving the request packet, demultiplexing the request, translating the request packet back into the original HTTP request, and passing the HTTP request to a local web server;

(e) in response to the peer node receiving an HTTP response from the Web server, translating the HTTP response into a response packet, and sending the response packet to the proxy server over the outbound socket connection; and

(f) in response to the proxy server receiving the response packet from the peer server, translating the response packet back into the HTTP response, and sending the HTTP response to the requesting Web browser.